NAT Support for Multiple Pools Using Route Maps

Document ID: 13739

Contents

Introduction

This document explains how the use of access lists versus route maps changes the functionality of Network Address Translation (NAT). For more information on NAT, refer to Cisco IOS NAT.

Prerequisites

Requirements

There are no specific requirements for this document.

Components Used

The information in this document is based on these software and hardware versions:

- Cisco 2500 Series Routers.
- Cisco IOS® Software Release 12.3(3).

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, make sure that you understand the potential impact of any command.

Conventions

Refer to Cisco Technical Tips Conventions for more information on document conventions.

Background Information

NAT **only** uses access lists and route maps when it needs to create a translation entry. If a translation entry already exists that matches the traffic then the translation entry will be used; any access lists or route maps will **not** be consulted. The difference between using an access list or route map is the type of translation entry that will be created.

Route Maps

When NAT uses a route map to decide to create a translation entry, it will always create a "fully extended" translation entry. This translation entry will contain both the inside and outside (local and global) address entries and any TCP or UDP port information. Refer to NAT: Local and Global Definitions for more information on inside and outside (local and global) addresses.

Access Lists (no overload)

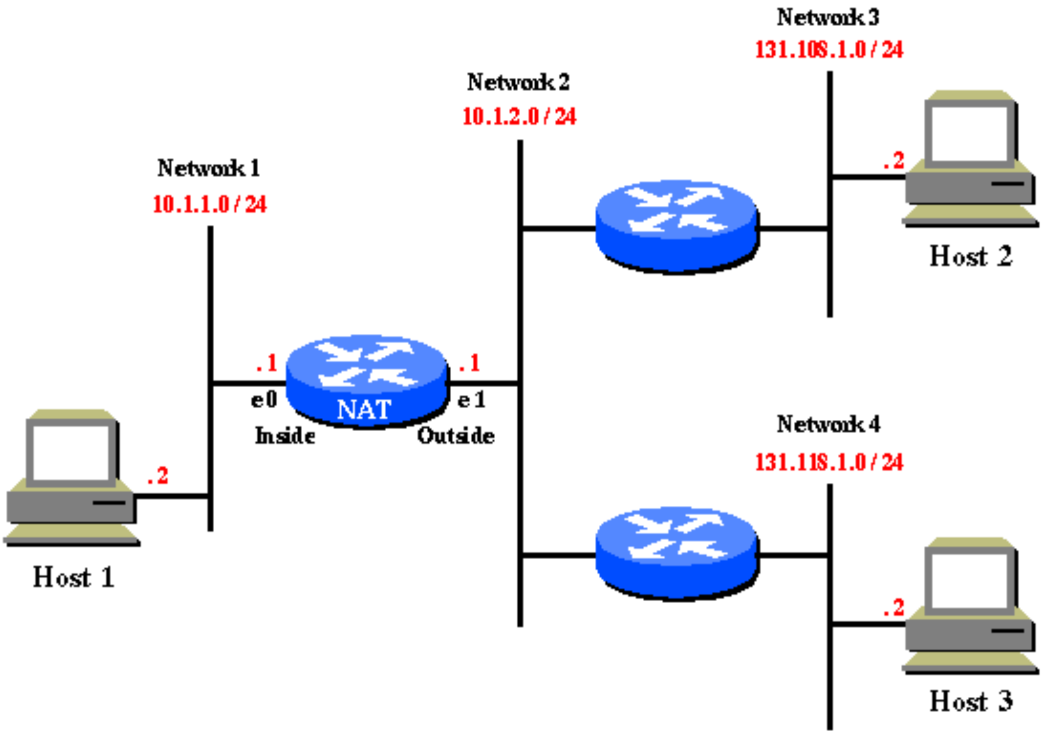
When NAT uses an access list to decide to create a translation entry, it will create a "simple" translation entry. This "simple" entry will only contain local and global IP address entries for just the inside or outside depending on whether the **ip nat inside** or **ip nat outside** command is configured. Also, it will not include any TCP or UDP port information.

Access Lists (with overload)

When NAT uses an access list, and overload has also been specified, NAT will create a "fully extended" translation entry. (See Note1). The operation is similar to the route−map case except that route−map has some additional features. See Note 2 for more details. You can see an example of a simple NAT translation entry and a fully extended NAT translation entry by selecting one of these links:

- Simple NAT translation entry
- Fully extended NAT translation entry

This is an example network diagram used to illustrate the difference between using a route map and an access list with NAT:



In this example network diagram, it is required that hosts on 10.1.1.0 be translated to the following:

- 131.108.2.0 when going to 131.108.1.0
- 131.118.2.0 when going to 131.118.1.0

Access List Approach

With an access list approach, you would do the following to translate the hosts on 10.1.1.0:

```
ip nat pool pool108 131.108.2.1 131.108.2.254 prefix-length 24

!--- Defines a pool of global addresses to be allocated as needed.

     ip nat pool pool118 131.118.2.1 131.118.2.254 prefix-length 24

     ip nat inside source list 108 pool pool108

!--- Establishes dynamic source translation, specifying the
     !--- access list defined below.

     ip nat inside source list 118 pool pool118

     interface ethernet0
       ip address 10.1.1.1 255.255.255.0
       ip nat inside

!--- Marks the interface as connected to the inside.

     interface ethernet1
       ip address 10.1.2.1 255.255.255.0
       ip nat outside

!--- Marks the interface as connected to the outside.


     access-list 108 permit ip 10.1.1.0 0.0.0.255 131.108.1.0 0.0.0.255

!--- Defines the access-list mentioning those addresses
     !--- that are to be translated.

     access-list 118 permit ip 10.1.1.0 0.0.0.255 131.118.1.0 0.0.0.255
```

Refer to IP Addressing and Services Commands for more information on these commands.

Host 1 to Host 2

Here is what happens when Host 1 Telnets to Host 2.

```
Packet on (Network 1) s:10.1.1.2(1024)     d:131.108.1.2(23)
     Packet on (Network 2) s:131.108.2.1(1024)  d:131.108.1.2(23)   (after NAT)
```

Because an access list was used by NAT to match this traffic a simple translation entry is created, which only includes inside translation information and no protocol or port information:

```
inside                         outside
         local        global          global         local
        10.1.1.2     131.108.2.1       ----           ----
```

Return packet: Host 2 to Host 1:

```
Packet on (Network 2)  s:131.108.1.2(23)  d:131.108.2.1(1024)
     Packet on (Network 1)  s:131.108.1.2(23)  d:10.1.1.2(1024)      (after NAT)
```

Host 1 to Host 3

With the above simple translation in place, here is what happens when Host 1 also Telnets to Host 3:

```
Packet on (Network 1)  s:10.1.1.2(1025)     d:131.118.1.2(23)
     Packet on (Network 2)  s:131.108.2.1(1025)  d:131.118.1.2(23)   (after NAT)
```

You can see that there is a problem. Packets going from 10.1.1.0 hosts to 131.118.1.0 hosts should get translated into 131.118.2.0, **not** 131.108.2.0. The reason that this happens is because there is already a NAT translation entry for 10.1.1.2 <−−> 131.108.2.1 which also matches the traffic between Host 1 and Host 3. Therefore, this translation entry will be used and access lists 108 and 118 are not checked.

While the simple translation entry is in place in the NAT translation table, it can be used by **any** outside user on any outside host to send a packet to Host 1 as long as the outside user uses the inside global address (131.108.2.1) for Host 1. Normally a static NAT translation is needed to allow this.

Route Map Approach

The correct way to configure the example in this document is to use route maps. With a route map approach, you would do the following to translate the hosts on 10.1.1.0:

```
ip nat pool pool−108 131.108.2.1 131.108.2.254 prefix−length 24
     ip nat pool pool−118 131.118.2.1 131.118.2.254 prefix−length 24

     ip nat inside source route−map MAP−108 pool pool−108

!−−− Establishes dynamic source translation, specifying
     !−−− the route−map MAP−108 which is defined below.

     ip nat inside source route−map MAP−118 pool pool−118

     !−−− Establishes dynamic source translation, specifying the route−map MAP−118.
     !−−− Here, the route−maps are consulted instead of
     !−−− access−lists (as in the previous case).



     interface ethernet0
       ip address 10.1.1.1 255.255.255.0
       ip nat inside
     interface ethernet1
       ip address 10.1.2.1 255.255.255.0
       ip nat outside

     access−list 108 permit ip 10.1.1.0 0.0.0.255 131.108.1.0 0.0.0.255
     access−list 118 permit ip 10.1.1.0 0.0.0.255 131.118.1.0 0.0.0.255

     route−map MAP−108 permit 10

!−−− Defines the Route−map MAP−108.

     match ip address 108

!−−− Specifies the criteria for translation. Here, the IP
     !−−− address mentioned in the access−list 108 is translated.
     !−−− The translation is defined in the
     !−−− ip nat inside source route−map MAP−108 pool pool−108 command.
     route-map MAP-118 permit 10

!--- Defines the Route-map MAP-108.

     match ip address 118

!--- The IP address mentioned in the access-list 118 is translated.
     !--- The translation is defined in the
     !--- ip nat inside source route-map MAP-118 pool pool-118 command.
```

Refer to IP Addressing and Services Commands for more information on these commands.

Host 1 to Host 2

Here is what happens when Host 1 Telnets to Host 2:

```
Packet on (Network 1) s:10.1.1.2(1024)     d:131.108.1.2(23)
     Packet on (Network 2) s:131.108.2.1(1024)  d:131.108.1.2(23)   (after NAT)
```

In this case, because a route map was used by NAT to match the traffic to be translated, NAT will create a fully extended translation entry, which includes both inside and outside translation information:

```
 inside                                outside
         local           global             global            local
     10.1.1.2:1024    131.108.2.1:1024   131.108.1.2:23     131.108.1.2:23
```

Return packet: Host 2 to Host 1:

```
Packet on (Network 2) s:131.108.1.2(23)  d:131.108.2.1(1024)
     Packet on (Network 1) s:131.108.1.2(23)  d:10.1.1.2(1024)      (after NAT)
```

Host 1 to Host 3

Now when Host 1 sends a packet to Host 3, this is what appears:

```
Packet on (Network 1) s:10.1.1.2(1025)     d:131.118.1.2(23)
     Packet on (Network 2) s:131.118.2.1(1025)  d:131.118.1.2(23)   (after NAT)
```

The translation worked correctly because the packet on (N1) does not match the fully extended translation entry that was used for the Host 1 to Host 2 traffic. Because the existing translation does not match, NAT creates another translation entry for the Host 1 to Host 3 traffic.

These are the fully extended translation entries on the NAT router:

```
inside                                 outside
         local           global             global            local
     10.1.1.2:1024    131.108.2.1:1024   131.108.1.2:23     131.108.1.2:23
     10.1.1.2:1025    131.118.2.1:1025   131.118.1.2:23     131.118.1.2:23
```

Because the NAT translation table has two full entries, it will correctly translate traffic going to the two different destinations from the same source.

Unlike the simple translation entry that was created via the access list, the fully extended translation entry created via the route map cannot be used by any other outside user to send a packet to Host 1. A static NAT translation is needed to allow this.

Note 1

In the case of access−list with overload, the configuration is similar to the access−list without overload case. The exception is that you need to add the keyword **overload** to the command **ip nat inside source list 108 pool pool108 and ip nat inside source list 118 pool pool118**.

Note 2

The advantage of using route−maps is that under the **match** command you can have more options other than source IP address. For example, under the route−map, **match interface** or **match ip next−hop** can be specified. By using route−maps, you can specify the IP address as well as the interface or the next−hop address to which the packet is to be forwarded. Therefore, route−maps with NAT are used in a scenario where the subscriber is multi−homing to different ISPs.

Related Information

- **NAT Ability to Use Route Maps with Static Translations**
- **Cisco IOS Network Address Translation**
- **Configuring Network Address Translation**
- **NAT: Local and Global Definitions**
- **Cisco IOS IP Command Reference for Addressing and Services, Release 12.3**
- **Technical Support & Documentation − Cisco Systems**

Contacts & Feedback | Help | Site Map

Updated: Aug 10, 2005 Document ID: 13739